# [WiP] Decision Log - Materials dashboard

| | |
|---|---|
| **Due Date** | |
| **Stakeholders** | juan.montilla@syensqo.com<br><br>mark.kwasnik@syensqo.com |
| **Selected Alternative** | |
| **Contributors** | |
| **Responsible** | OLIVEIRA, Tiago |



## High Level Design

*Tbd*

**Application - Workflow**

## Low Level Design

Production

*TBD*

Quality

*TBD*

## Risks

## Non-Risks

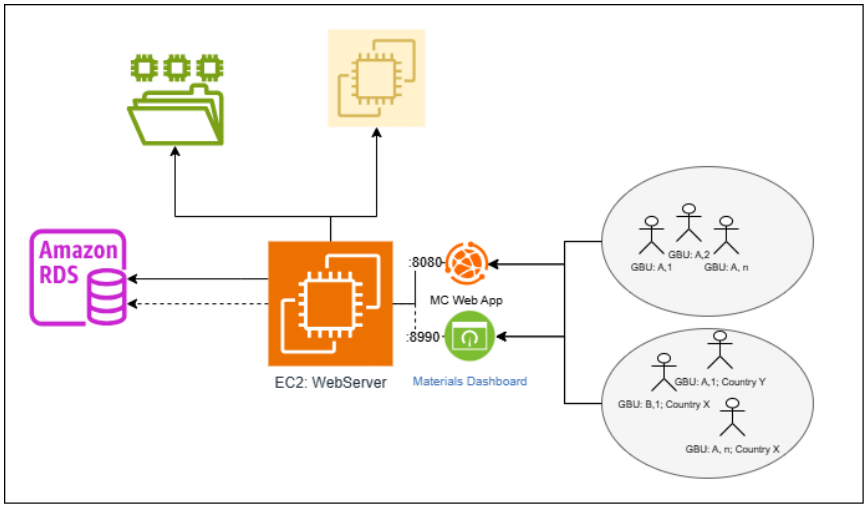
## Tradeoff Analysis

### **Alternative 1** - Integrate Dashboard into MaterialCenter Application

Embed the Streamlit dashboard directly within the existing MaterialCenter application, utilizing its current data model and infrastructure. The dashboard becomes an additional page or feature within the core system.

**Tradeoff Analysis:**

- **Cost Efficiency:** Low incremental cost, as it reuses existing infrastructure.
- **Complexity:** Minimal, since it leverages current setup and data sources.
- **Principles & Standards:** High alignment with established practices; ensures consistency.
- **Maturity & Operations:** Mature operational model; maintenance and support are streamlined but may add load to the core app.

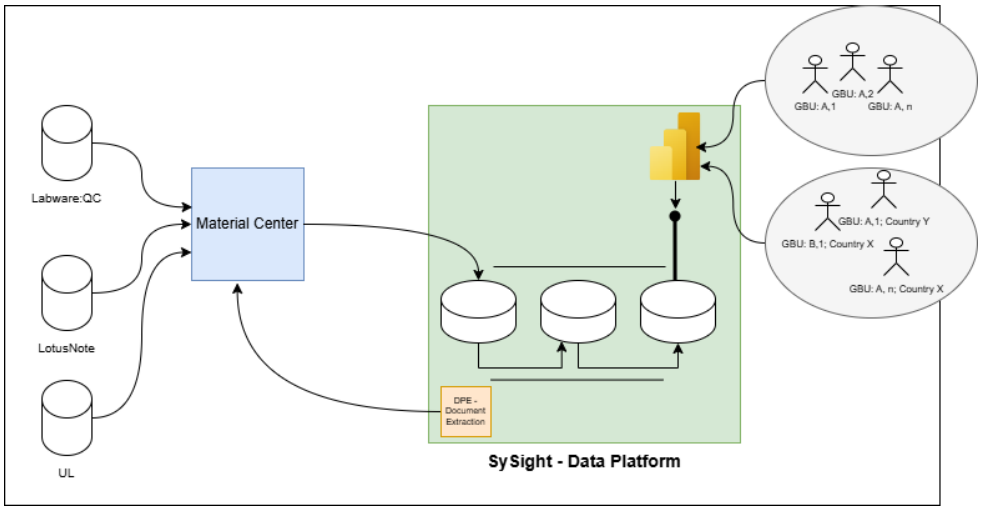



## **Alternative 2** - Deploy on SySight Data Platform

Host the dashboard on SySight, Syensqo's data platform, utilizing standard technologies such as data pipelines for ingestion, medallion architecture for data management, and GraphQL APIs. Visualization could be via Streamlit or PowerBI.

**Tradeoff Analysis:**

- **Cost Efficiency:** Potentially higher, but benefits from shared platform resources.
- **Complexity:** Moderate to high, due to integration with platform standards and data pipelines.
- **Principles & Standards:** Strong alignment with enterprise data architecture and analytics best practices.
- **Maturity & Operations:** High maturity; robust governance and scalability, but onboarding may require more effort.




**Alternative 3** - Lightweight Deployment via AWS Lambda (PaaS/FaaS)

Deploy the dashboard as a lightweight, standalone application using AWS Lambda or similar serverless services, minimizing operational overhead and infrastructure management - ideal for temporary or low-traffic use.

**Tradeoff Analysis:**

- **Cost Efficiency:** Very cost-effective for low usage; pay-per-use model.
- **Complexity:** Low, with minimal setup and maintenance.
- **Principles & Standards:** May not fully align with enterprise standards for long-term solutions.
- **Maturity & Operations:** Limited operational complexity; best suited for temporary or pilot deployments.

**Alternative 4 -** Deploy in Shared Application Cluster

Integrate the dashboard into an existing shared cluster that hosts multiple web applications, allowing resource sharing and centralized management.

**Tradeoff Analysis:**

- **Cost Efficiency:** Efficient use of shared resources; moderate cost.
- **Complexity:** Moderate, as it requires integration with cluster management and existing apps.
- **Principles & Standards:** Good alignment with shared services approach.
- **Maturity & Operations:** Mature operational model; may face resource contention or dependency issues.

## Evaluation Matrix

| | **Architectural Approach** | | | | | |
|---|---|---|---|---|---|---|
| 1 | **Alternative /Evaluation Vector** | | **Alternative 1** - Integrate Dashboard into MaterialCenter Application | **Alternative 2** - Deploy on SySight Data Platform | **Alternative 3** - Lightweight Deployment via AWS Lambda (PaaS /FaaS) | **Alternative 4 -** Deploy in Shared Application Cluster |
| 2 | **Technical Debts** | Paid | | | | |
| | | Added | | | | |
| 3 | **Major Work Block** | | | | | |
| 4 | **Performance** | | | | | |
| 5 | **Scalability** | | | | | |
| 6 | **Consistency** | | | | | |
| 7 | **Resource - Cost Efficiency** | | | | | |

## References